var width = 960 / 10,

height = 600 / 10,

marginTop = 20;

var pad02 = d3.format("02d"),

formatMonth = d3.time.format("%B");

var label = d3.select("#month");

var projection = d3.geo.albers()

.scale(width \* 1.38)

.translate([width \* .48, height \* .52]);

var color = d3.scale.linear()

.domain([-4, 0, 4])

.range(["#7c003d", "#ffffbf", "#288219"])

.interpolate(d3.interpolateHcl);

d3.json("climate-divisions.json", function(collection) {

// Precompute the projected coordinates.

collection.features.forEach(function(feature) {

var coordinates = feature.geometry.coordinates;

coordinates[0] = coordinates[0].map(projection);

});

d3.tsv("pdsi.json", function(pdsi) {

var pdsiById = {};

pdsi.forEach(function(d) { pdsiById[d.id] = d; });

var year = d3.select("#chart").selectAll(".year")

.data(d3.range(1895, 2013))

.enter().append("div")

.attr("class", "year")

.style("top", function(year) { return (201 - Math.floor(year / 10)) \* (height + marginTop) + marginTop + "px"; })

.style("left", function(year) { return (year % 10) \* width + "px"; });

year.append("span")

.attr("class", "label")

.classed("decade", function(year) { return !(year % 10); })

.text(String);

var canvas = year.select(function() { return this.appendChild(newCanvas()); })

.datum(function(year) {

var canvasByMonth = {};

return {

context: this.getContext("2d"),

canvasByMonth: function(month) {

var canvas = canvasByMonth[month];

if (!canvas) {

canvasByMonth[month] = canvas = newCanvas();

var context = canvas.getContext("2d");

collection.features.forEach(function(feature) {

var z = pdsiById[feature.id][year + pad02(month) + "01"];

if (!isNaN(z)) {

context.fillStyle = color(z);

pathFeature(context, feature);

context.fill();

}

});

}

return canvas;

}

};

});

var month = 0;

setInterval(function() {

if (++month > 12) month = 1;

canvas.each(function(d) { d.context.drawImage(d.canvasByMonth(month), 0, 0, width, height); });

label.text(formatMonth(new Date(2012, month - 1, 1)));

}, 1000 / 12);

});

function newCanvas() {

var canvas = document.createElement("canvas"),

context = canvas.getContext("2d"),

ratio = (window.devicePixelRatio / context.webkitBackingStorePixelRatio) || 1;

d3.select(canvas)

.attr("width", width \* ratio)

.attr("height", height \* ratio)

.style("width", width + "px")

.style("height", height + "px");

context.scale(ratio, ratio);

context.clearRect(0, 0, width, height);

return canvas;

}

// This only works for simple polygons with no holes.

function pathFeature(context, feature) {

var coordinates = feature.geometry.coordinates[0],

i = 0,

n = coordinates.length;

context.beginPath();

context.moveTo(coordinates[0][0], coordinates[0][1]);

while (++i < n) context.lineTo(coordinates[i][0], coordinates[i][1]);

context.closePath();

}

});